

BCM5880 Cryptographic Module
Security Policy
Document *Version 1.1*

Broadcom Corporation

Revision Date: 2010-02-02

TABLE OF CONTENTS

1. MODULE OVERVIEW3

2. SECURITY LEVEL.....5

3. MODES OF OPERATION.....5

4. PORTS AND INTERFACES7

5. IDENTIFICATION AND AUTHENTICATION POLICY.....10

6. ACCESS CONTROL POLICY.....11

 DEFINITION OF SERVICES.....11

 DEFINITION OF CRITICAL SECURITY PARAMETERS (CSPs).....15

 DEFINITION OF CSPs MODES OF ACCESS20

7. OPERATIONAL ENVIRONMENT.....23

8. SECURITY RULES23

9. PHYSICAL SECURITY POLICY26

 PHYSICAL SECURITY MECHANISMS26

10. MITIGATION OF OTHER ATTACKS POLICY.....26

11. REFERENCES26

12. DEFINITIONS AND ACRONYMS.....27

1. Module Overview

The BCM5880 Cryptographic Module (HW P/N BCM5880KFBG, Version C0; FW Version C0; SW Version R0), a single-chip encased in hard opaque tamper evident IC packaging, is a highly integrated system on a chip. The BCM5880 Cryptographic Module's security architecture includes an innovative virtualized Secure/Open Domain execution environment. This innovative environment provides different software execution contexts that allow additional logical separation of the cryptographic operations from IO related operations. For the purpose of FIPS 140-2 validation the physical boundary of the chip is used as the security boundary of the cryptographic module (note that no additional FIPS security claim is made for the different logical software execution contexts located within the physical boundary).

The BCM5880 Cryptographic Module's FIPS 140-2 boundary is defined as:

- The external surface of the BCM5880 chip including the hard, opaque encapsulating material that physically protects all module components.

The figures below picture the cryptographic module's physical boundary, interfaces, and logical software execution contexts within the physical boundary.

Figure 1 – Image of the Cryptographic Module Physical Boundary



Figure 1.a: BCM5880 (Top)

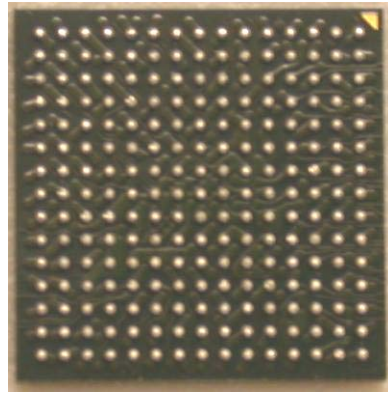
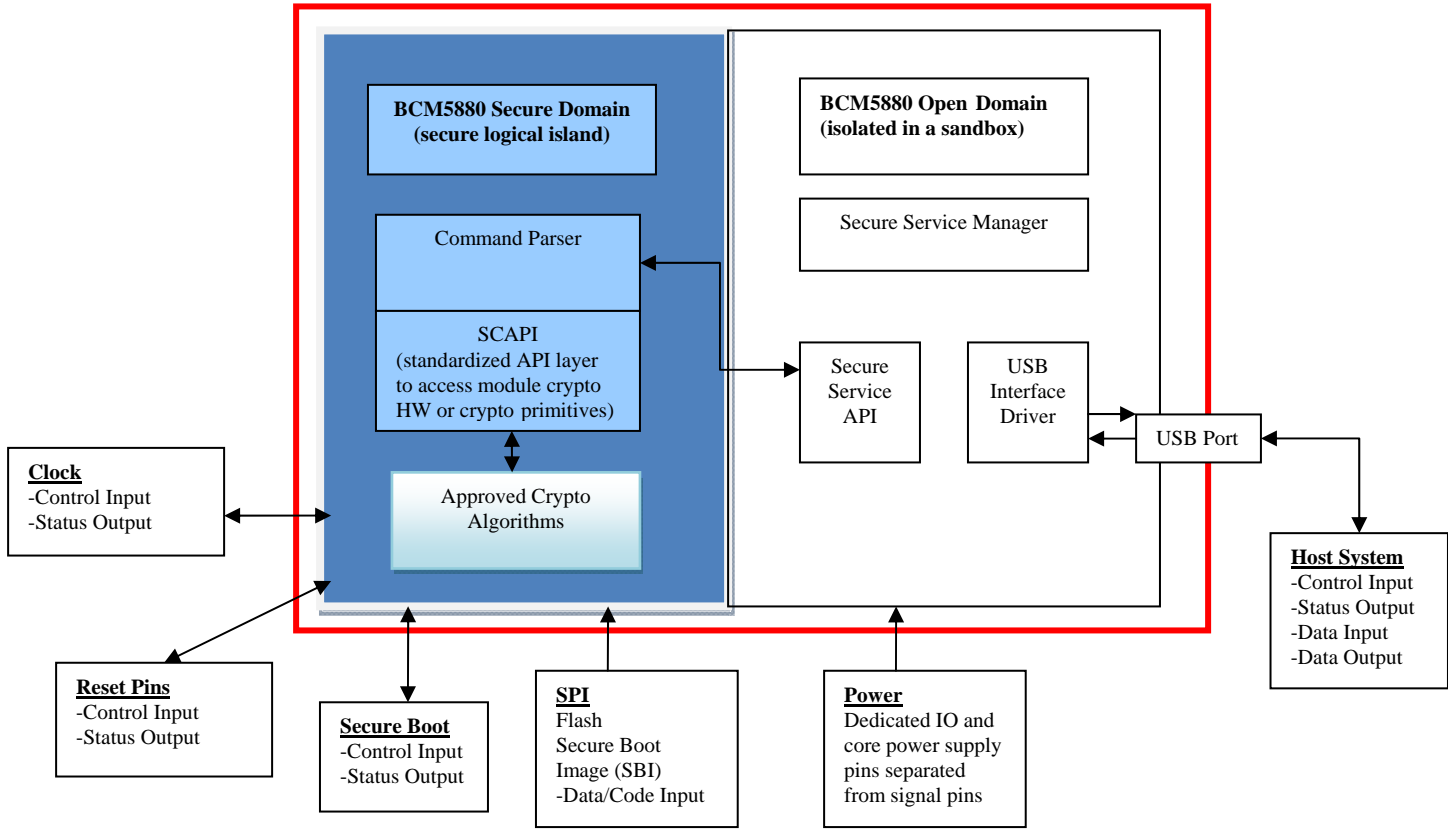


Figure 1.b: BCM5880 (Bottom)

Figure 2 – Block Diagram of module Interfaces & logical Software Execution Contexts



2. Security Level

The cryptographic module meets the overall requirements applicable to Level 3 security of FIPS 140-2.

Table 1 – Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	3
Module Ports and Interfaces	3
Roles, Services, and Authentication	3
Finite State Model	3
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	N/A

3. Modes of Operation

FIPS Approved mode of operation

The BCM5880 Cryptographic Module is configured to run in FIPS Approved mode of operation when the external SECURE_BOOT: CONTROL INPUT: SECURE_BOOT pin is set high.

The BCM5880 Cryptographic Module supports a single FIPS Approved mode of operation. The user can determine that the cryptographic module is running in FIPS Approved mode of operation when the SECURE_BOOT: STATUS OUTPUT: TEST_SEC_BOOT pin is high.

Non-FIPS Approved mode of operation

The BCM5880 Cryptographic Module is configured to run in non-FIPS Approved mode of operation when the external SECURE_BOOT: CONTROL INPUT: SECURE_BOOT pin is set low.

The BCM5880 Cryptographic Module supports a single non-FIPS Approved mode of operation, Software Development Test Mode. The user can determine that the cryptographic module is running in non-FIPS Approved mode of operation when the SECURE_BOOT: STATUS OUTPUT: TEST_SEC_BOOT pin is low.

In order to switch between FIPS Approved mode of operation and non-FIPS Approved mode of operation the module must be power-cycled. Power-cycling zeroizes all volatile plaintext critical security parameters. Module HW\FW\SW enforces that non-volatile plaintext critical security parameters cannot be shared, used, or viewed between modes of operation.

When the module is configured to run in the non-FIPS Approved mode of operation no claim is made for any cryptographic operation.

Approved and Allowed Algorithms

The module implements the following Approved and allowed cryptographic algorithms using a hardware crypto engine called [SMAU - Crypto/Auth] block. This block is instantiated twice in the Secure Memory Access Unit or SMAU. One instance is being used for offloading generic cryptographic operations. The other instance is being used to support secure caching of instruction and data stored externally in encrypted and integrity-protected format. Individual self-tests are conducted after power-on to test the two instantiations independently. Each algorithm implementation is used during different scenarios. They are never used simultaneously for the same operation. Each algorithm implementation has its own algorithm certificate and has its own power on self-test.

<u>AES</u> :	[SMAU – Crypto/Auth] block ECB, CBC, CTR, CMAC 128, 192, 256 keys	Certificate #1070
<u>CCM</u> :	[SMAU – Crypto/Auth] block 128 key size, Nonce Len 12, Tag Len 4, 8, 12, 16	Certificate #1070
<u>HMAC-SHA-1 & HMAC-SHA-256</u> :	[SMAU – Crypto/Auth] block	Certificate #602
<u>SHA-1 & SHA-256</u> :	[SMAU – Crypto/Auth] block	Certificate #1011
<u>PRNG</u> :	FIPS 186-2 Appendix 3.1 X-orig and 3.2 K-orig SHA-1 based	Certificate #605
<u>ECDSA</u> :	Signature generation, signature verification 256-bit key,	Certificate #128
<u>DSA</u> :	Signature generation, signature verification 1024-bit key,	Certificate #354
<u>RSA</u> :	Signature generation, signature verification 2048-bit keys	Certificate #507

Non-Approved Algorithms

The module implements the following non-Approved cryptographic algorithms

EC Diffie-Hellman: Module functions implement Diffie-Hellman primitives used for key agreement as allowed by FIPS 140-2 Implementation Guidance 7.1. The functions are implemented following the SP800-56A standard.

NDRNG: Internal module source utilizing free running oscillators to capture thermal noise as the source of randomness. The NDRNG is used to collect entropy to be fed to the FIPS 186-2 PRNG.

Note that 135MB of entropy data have been tested for strength via MinEntropy and STS testing. The test results reported 7.8952 bits per byte strength.

Strength of key generation:

- Seed Key:
 - 505 bits of strength:
 - (Entropy 7.8952 bits per byte strength) X (512 bit length of NDRNG output).
- Strongest key the BCM5880 Cryptographic Module will generate:
 - 256 bits of strength.

Key Establishment

EC Diffie-Hellman allowed as per FIPS 140-2 Implementation Guidance 7.1.

- **Strength of key establishment: $K_{ECDH-PRIV}$**
 - 256 bit random number used for ECDH key exchange. Strength = 128 bit since the derived session key is 128-bit.
 - Ephemeral key generated based on FIPS186-2 Appendix 3.1 PRNG algorithm during the ECDH session establishment. It is erased after the symmetrical session key is derived.
- **K_{SS} 128-bit AES-CCM mode key**
 - Session key derived during the EC Diffie-Hellman Key Exchange service. Module will use this key for secure communications to/from host system.
 - Derived during the EC Diffie-Hellman Key Exchange service via SHA256-based KDF function

4. Ports and Interfaces

The BCM5880 Cryptographic Module provides physical ports as listed in Table 2 below.

Table 2 – Physical Ports

Note: The BCM5880 chip has a total of 87 pins. Each BCM5880 Interface Group listed in Table 2 contains several BCM5880 pins. Unused Interface Groups will be marked as “Non-Available” because they are currently disabled by the cryptographic module.

Clock group:	Control Input	- Clock. - Clock Output Enabled.
	Status Output	- Clock Output.
Reset group:	Control Input	- Resets. - Indication that the system power supply is stable.

disabled <u>and</u> logic is put in reset state.	Intended use in the future: Data Input Data Output	Intended use in the future: Data received or transmitted For contacted Smart Card application
UART group: Clock to the group block is disabled <u>and</u> logic is put in reset state.	Non-Available Intended use in the future: Data Input Data Output	Non-Available Intended use in the future: Data received or transmitted for UART console application
LPC group: Clock to the group block is disabled <u>and</u> logic is put in reset state.	Non-Available Intended use in the future: Data Input Data Output Control Input Status Output	Non-Available Intended use in the future: Data, Control or Status information exchanged for TPM application
JTAG group: Completely disabled by HW in FIPS mode. Module HW\FW\SW enforces that non-volatile plaintext critical security parameters cannot be shared, used, or viewed in Non-FIPS Approved mode.	Non-Available	Non-Available
Test Control group: Completely disabled in FIPS mode. Used during manufacturing to test for defects. Module HW\FW\SW enforces that non-volatile plaintext critical security parameters cannot be shared, used, or viewed in Non-FIPS Approved mode.	Non-Available	Non-Available
Power group	Power is distributed to the chip using designated IO and core power pins that are completely separated from any signal pin groups. Power pins are only	

	connected to the internal power planes of the silicon chip.	
--	---	--

5. Identification and Authentication Policy

Assumption of Roles

The BCM5880 Cryptographic Module supports two operator roles, User and Cryptographic-Officer. The cryptographic module implements identity-based operator authentication.

Authentication is accomplished via a 256-bit ECDSA-based signature verification process. A single 256-bit ECDSA public key is embedded in the module's SBI during manufacturing (Secure Boot Image: an authenticated software extension of the module's BOOT ROM. SBI software is part of the BCM5880 Cryptographic Module). The 256-bit ECDSA public key is used to authenticate the operator during the establishment of an ECDH secure session between the module and the operator on the external host system.

After an operator is authenticated successfully, the operator can assume either the role of the Cryptographic Officer or the role of the User. The module allows the operator to perform both CO and User services.

Table 3 - Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
User	Identity-based operator authentication	<ul style="list-style-type: none"> 256-bit ECDSA signature verification
Cryptographic-Officer	Identity-based operator authentication	<ul style="list-style-type: none"> 256-bit ECDSA signature verification

Table 4 – Strengths of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
ECDSA Signature Verification (256 bit)	<p>The probability that a random attempt will succeed or a false acceptance will occur is $1/2^{128}$ which is less than 1/1,000,000.</p> <p>The probability of successfully authenticating to the module within one minute is $3,750/2^{128}$ which is less than 1/100,000. The module will only allow one attempt to verify the operator – if that attempt fails the module will be in an error</p>

	state and must be rebooted to try and become operational again. Please see section “8. Security Rules” below (security rules imposed by the vendor) for the detail supporting this calculation.
--	---

6. Access Control Policy

Definition of Services

The cryptographic module supports the following authenticated services defined in Table 5:

Table 5 – Authenticated Services

Name of Service	Description of Service
Generate Key	This service generates an AES or HMAC key to be used during operator requested services.
AES Encrypt	This service encrypts bulk operator supplied data using a previously generated AES key.
AES Decrypt	This service decrypts bulk operator supplied data using a previously generated AES key.
SHA-1 Hashing	This service generates a SHA-1 digest on operator supplied data.
SHA-256 Hashing	This service generates a SHA-256 digest on supplied data.
Load Key	<p>This service allows an operator to load a key into the module’s key cache.</p> <p>The key being loaded can be a private key or a public key of an asymmetrical key pair, or a symmetrical key for AES or HMAC.</p> <p>All keys loaded via this service are being protected by the ECDH secure session via 128-bit AES-CCM encryption and integrity protection.</p>
RSA Signature Verification	This service performs RSA Signature Verification on operator supplied data with a previously loaded public key (see service

	“Load Key”).
DSA Signature Verification	This service performs DSA Signature Verification on operator supplied data with a previously loaded public key (see service “Load Key”).
ECDSA Signature Verification	This service performs ECDSA Signature Verification on operator supplied data with a previously loaded public key (see service “Load Key”).
RSA Signature Generation	This service performs RSA Signature Generation on operator supplied data with a previously loaded private key (see service “Load Key”).
DSA Signature Generation	This service performs DSA Signature Generation on operator supplied data with a previously loaded private key (see service “Load Key”).
ECDSA Signature Generation	This service performs ECDSA Signature Generation on operator supplied data with a previously loaded private key (see service “Load Key”).
Generate Random Number	This service generates a random number with the module’s FIPS 186-2 PRNG and outputs the generated random number to the requesting operator.
EC Diffie-Hellman Key Exchange	This service is comprised of several steps which establish a session key between the module and an external entity.
HMAC Request	Compute an HMAC on an operator supplied blob of data with a previously generated or loaded secret key.

The cryptographic module supports the following unauthenticated services defined in Table 6:

Table 6 – Unauthenticated Services

Name of Service	Description of Service
Self Test	This service executes the suite of self-tests required by FIPS 140-2. Self-tests are invoked by power cycling the module.
Show Status	This service provides the current status of the cryptographic module.
Get Info	This service computes and outputs the ECDSA device public key of the cryptographic module
Get Version	This service returns the version/revision information of the cryptographic module
Zeroize	<ul style="list-style-type: none"> Power-cycle or hard reset will zeroize all volatile critical security parameters including internally generated CSPs or loaded keys. When the ZEROIZE PIN within the Secure Boot group physical interface is turned high all volatile and non-volatile plaintext critical security parameters will be zeroized – after this the module will not boot again.

Table 7 – Specification of Service Inputs & Outputs

Service	Control Input	Data Input	Data Output	Status Output
Generate Key	Key Type	N/A	Key Handle	Success/fail
AES Encrypt	Length Key Handle	Plaintext	Ciphertext	Success/fail
AES Decrypt	Length Key Handle	Ciphertext	Plaintext	Success/fail
SHA-1 Hashing	Hash Type	Data Blob	Digest	Success/fail
SHA-256 Hashing	Hash Type	Data Blob	Digest	Success/fail
Load Key	Key Type Key Handle	Key	N/A	Success/fail
RSA Signature Verification	Hash Length Key Handle	Hash Blob Signature	N/A	Success/fail

Service	Control Input	Data Input	Data Output	Status Output
DSA Signature Verification	Hash Length Key Handle	Hash Blob Signature	N/A	Success/fail
ECDSA Signature Verification	Hash Length Key Handle	Hash Blob Signature	N/A	Success/fail
RSA Signature Generation	Hash Length Key Handle	Hash Blob	Signature	Success/fail
DSA Signature Generation	Hash Length Key Handle	Hash Blob	Signature	Success/fail
ECDSA Signature Generation	Hash Length Key Handle	Hash Blob	Signature	Success/fail
Generate Random Number	PRNG Type Length	N/A	Random Number	Success/fail
EC Diffie-Hellman Key Exchange (comprised of two steps)	Header info.	EC Diffie-Hellman key establishment data received from Host System	EC Diffie-Hellman key establishment data sent to Host System	Success/fail
HMAC Request	Length Hash Type Key Handle	Data Blob	MAC	Success/fail
Self Test	N/A (Power cycle)	N/A	N/A	Success/fail
Show Status	N/A	N/A	N/A	All the above Status Output (Table 7 Specification of Service Inputs & Outputs) Status Output of Interface groups (Table 2 Physical Ports)
Get Info	N/A	N/A	Cryptographic Module device public key KDI-EC-PUB	Success/fail

Service	Control Input	Data Input	Data Output	Status Output
Get Version	N/A	N/A	Version and Revision information of the Cryptographic Module	Success/fail
Zeroize	N/A	N/A	N/A	N/A

Definition of Critical Security Parameters (CSPs)

The following are the CSPs contained in the module.

Table 8 – Secret and Private Keys

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
K_{ECDH-PRIV} 256 bit random number used for ephemeral ECDH key.	Used to establish an ECDH based session key.	Ephemeral key generated internally via PRNG per FIPS 186-2, appendix 3.1.	Stored in plaintext internally in the module's [Scratch RAM] block. Key-to-entity association: Associated with a key index = 1 in OTP.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Additionally always destroyed after the symmetrical session key is established.
K_{AES} 128 bit AES key. A unique value for each module.	Used to encrypt and decrypt the Secure Boot Image (SBI) when the SBI is loaded	Generated internally during manufacturing via PRNG per FIPS 186-2, appendix 3.1.	Stored in plaintext internally in OTP. When in use it is temporality copied to the [Scratch RAM] block. Key-to-entity association: Key index = 2 in OTP.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
K_{HMAC} 256 bit HMAC-SHA-256 key. A unique value for each module.	Used to protect and verify the SBI.	Generated internally during manufacturing via PRNG per FIPS 186-2, appendix 3.1.	Stored in plaintext internally in OTP. When in use it is temporality copied to the [Scratch RAM] block. Key-to-entity association: Key index = 3 in OTP.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
K_{DI-EC-PRIV} 256 bit ECDSA private key. A unique value for each module.	Used to establish the mutually authenticated ECDH secure session communication channel between the module and an external entity. Used as the identity key of the module in these authenticated communications.	Generated internally during manufacturing via PRNG per FIPS 186-2, appendix 3.1.	Stored in plaintext internally in OTP. When in use it is temporality copied to the [Scratch RAM] block. Key-to-entity association: Key index = 4 in OTP.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
K_{APP-AES} 128, 192 or 256 bit AES keys.	Used to encrypt/decrypt application data when external applications issue encrypt or decrypt service requests.	Generated (optional): internally during operation via PRNG per FIPS 186-2, appendix 3.1. See Generate Key service.	Stored in the volatile “key cache” within the [Scratch RAM] block. Key-to-entity association: “key cache” handle. Note this handle is given by the application that requested the creation of the key so that application can request encryption/decryption with the key at a later point in time.	Entry (optional): Entered into the module by Load Key service Entry Key-to-entity association: Session key derived during the EC Diffie-Hellman Key Exchange service. Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
K_{APP-HMAC} 160 bit HMAC keys (SHA-1). 256 bit HMAC keys (SHA-256).	Used to protect and verify application data when external applications issue protection or verification service requests.	Generated (optional) internally during operation via PRNG per FIPS 186-2, appendix 3.1. See Generate Key service.	Stored in the volatile “key cache” within the [Scratch RAM] block. Key-to-entity association: “key cache” handle. Note this handle is given by the application that requested the creation of the key so that application can request protection/ verification with the key at a later point in time.	Entry (optional): Entered into the module by Load Key service Entry Key-to-entity association: Session key derived during the EC Diffie-Hellman Key Exchange service. Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
K_{APP-PRIV} 1024 bit DSA 2048 bit RSA 256 bit ECDSA	Used to perform signature generation during the RSA, DSA or ECDSA Signature services.	N/A	Multiple instances. Stored in the volatile “key cache” within the [Scratch RAM] block. Key-to-entity association: “key cache” handle. Note this handle is given by the application that requested the entry of the key so that the application can request signature generation with the key at a later point in time.	Entry: Entered into the module by Load Key service Entry Key-to-entity association: This is a private key that is associated with the public key member of a key-pair. Output: N/A Output Key-to-entity association: N/A.	When the zeroize service is requested. Always destroyed after each reset cycle.

Key	Description/Usage	Generation	Storage	Entry/Output	Destruction
K_{ECDH-SS} 256 bit ephemeral ECDH shared secret.	Used to derive the session key K_{SS}	Derived using ECDH key exchange algorithm based on K_{ECDH-PRIV} and K_{ECDH-OP-PUB}	Stored only temporarily in the scratch RAM, erased after K_{SS} is derived Key-to-entity association: Associated with a session ID during the ECDH secure session establishment.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Additionally always destroyed after the symmetrical session key is established.
K_{SS} 128 bit AES key.	Session key derived during the EC Diffie-Hellman Key Exchange service. The module will use this key for secure communications to/from the external host system.	Generated during the EC Diffie-Hellman Key Exchange service via SHA256-based KDF function.	Stored in the volatile “key cache” within the [Scratch RAM] block. Key-to-entity association: Only one session key exists at any given point in time.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Zeroize service. Temporary copy in [Scratch RAM] block always destroyed after each reset cycle.
PRNG Seed Key 512 bits with 505 bits of entropy strength	Entropy value fed to the FIPS 186-2 PRNG.	Gathered from internal module NDRNG utilizing free running oscillators to capture thermal noise.	Generated via NDRNG and stored in PRNG registers Key-to-entity association: Only one PRNG seed key exists at any given point in time.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Reset PRNG or power cycle the chip.
PRNG State	State of the module’s FIPS 186-2 PRNG.	Generated within the module’s FIPS 186-2 PRNG.	Stored in PRNG registers. Key-to-entity association: The PRNG maintains one state at a given time.	Entry: N/A Entry Key-to-entity association: N/A Output: N/A Output Key-to-entity association: N/A.	Reset PRNG or power cycle the chip.

Definition of Public Keys:

The following are public keys contained in the module.

Table 9 – Public Keys

Key	Description/Usage	Generation	Storage	Entry/Output
K_{DI-EC-PUB} 256 bit ECDSA public key. A unique value for each module.	Used by the operator to authenticate the cryptographic module in a mutually authenticated secure session	Computed internally upon each get_info request per ECDSA algorithm	Stored only temporarily in the scratch RAM during the processing of the get_info service Key-to-entity association: Public part of the device identity key.	Entry: N/A Entry Key-to-entity association: N/A Output: as the result of get_info service Output Key-to-entity association: embedded in the get_info command response.
K_{ECDH-PUB} 256 bit public ephemeral ECDH key of the cryptographic module	Used to establish an ECDH based session.	Ephemeral public key generated internally based on ECDH algorithm defined in SP800-56A	Stored only stored temporarily in the scratch RAM during the process of establishing the ECDH session, erased after the session key is established Key-to-entity association: Public key of the ephemeral ECDH key pair.	Entry: N/A Entry Key-to-entity association: N/A Output: as the result of the ECDH key exchange Output Key-to-entity association: embedded in the command response for ECDH key exchange.
K_{ECDH-OP-PUB} 256 bit public ephemeral ECDH key of the operator	Used to establish an ECDH based session.	Ephemeral public key generated and signed by the operator, pass into the cryptographic module during ECDH session key exchange	Stored only stored temporarily in the scratch RAM during the process of establishing the ECDH session, erased after the session key is established Key-to-entity association: Associated with the authentication session. Only one session is active.	Entry: input of the ECDH key exchange Entry Key-to-entity association: embedded in the command for ECDH key exchange. Output: N/A Output Key-to-entity association: N/A

Key	Description/Usage	Generation	Storage	Entry/Output
K_{APP-PUB} 1024 bit DSA 2048 bit RSA 256 bit ECDSA	Used to perform signature verification during the RSA, DSA or ECDSA Signature Verification services.	N/A	Stored in the volatile “key cache” within the [Scratch RAM] block on the block diagram. Key-to-entity association: “key cache” handle. Note this handle is passed back to the application that requested the entry of the key so that the application can request signature verification with the key at a later point in time.	Entry: Entered into the module by the Load Key service. Entry Key-to-entity association: This is a public key that is associated with the private key member of a key-pair. Output: N/A Output Key-to-entity association: N/A.
K_{OP-PUB} 256 bit ECDSA	Operator’s public key Used to authenticate the operator during an ECDH secure session.	N/A	Stored in the on-chip RAM. Key-to-entity association: This key is located at a fixed offset of the SBI image known to the implementation of the cryptographic module.	Entry: Embedded in the SBI during the manufacturing process. Entry Key-to-entity association: This is a public key that is associated with the private key member of a key-pair. Output: N/A Output Key-to-entity association: N/A.

Definition of CSPs Modes of Access

Table 10 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as:

- **G = Generate:** The module generates the CSP.
- **R = Read:** The module reads the CSP. The read access is typically performed before the module uses the CSP.
- **W = Write:** The module writes the CSP. The write access is typically performed

after a CSP is imported into the module, or the module generates a CSP, or the module overwrites an existing CSP.

- **Z = Zeroize:** The module zeroizes the CSP.

Table 10 – CSP Access Rights within Roles & Services

Role		Service	Cryptographic Keys and CSPs Access Operation
C.O.	User		
X	X	Generate Key	G $K_{APP-AES}$ G $K_{APP-HMAC}$ For each service call a handle to the generated key will be passed back to the operator.
X	X	AES Encrypt	R $K_{APP-AES}$ For each service request the operator will indicate which $K_{APP-AES}$ key to use by passing in the key's handle as input.
X	X	AES Decrypt	R $K_{APP-AES}$ For each service request the operator will indicate which $K_{APP-AES}$ key to use by passing in the key's handle as input.
X	X	SHA-1 Hashing	N/A
X	X	SHA-256 Hashing	N/A
X	X	Load Key	W $K_{APP-PUB}$ W $K_{APP-AES}$ W $K_{APP-HMAC}$ For each service request a handle to the loaded key will be passed back to the operator.
X	X	RSA Signature Verification	R $K_{APP-PUB}$ For each service request the operator will indicate which $K_{APP-PUB}$ RSA key to use by passing in the key's handle as input.
X	X	DSA Signature Verification	R $K_{APP-PUB}$ For each service request the operator will indicate which $K_{APP-PUB}$ DSA key to use by passing in the key's handle as input.
X	X	ECDSA Signature	R $K_{APP-PUB}$ For each service request the operator will indicate which

		Verification	$K_{APP-PUB}$ ECDSA key to use by passing in the key's handle as input.
X	X	RSA Signature Generation	R $K_{APP-PRIV}$ For each service request the operator will indicate which $K_{APP-PRIV}$ RSA key to use by passing in the key's handle as input.
X	X	DSA Signature Generation	R $K_{APP-PRIV}$ For each service request the operator will indicate which $K_{APP-PRIV}$ DSA key to use by passing in the key's handle as input.
X	X	ECDSA Signature Generation	R $K_{APP-PRIV}$ For each service request the operator will indicate which $K_{APP-PRIV}$ ECDSA key to use by passing in the key's handle as input.
X	X	Generate Random Number	R PRNG Seed Key (note: a new Seed Key is generated for each call to service Generate Random Number). R PRNG State The PRNG is seeded with the PRNG Seed Key. The random number generated by the PRNG is returned to the operator requesting the service.
X	X	EC Diffie-Hellman Key Exchange	R $K_{DI-EC-PRIV}$ R $K_{ECDH-PRIV}$ R K_{OP-PUB} G $K_{ECDH-PUB}$ R $K_{ECDH-OP-PUB}$ G $K_{ECDH-SS}$ G K_{SS} The operator establishes a secure ECDH key exchange session with a derived session key K_{SS}
X	X	HMAC Request	R $K_{APP-HMAC}$ For each service request the operator will indicate which key to use by passing in key handles as input.
X	X	Self Test	N/A
X	X	Show Status	N/A
X	X	Gen Info	R $K_{DI-EC-PUB}$
X	X	Gen Version	N/A
X	X	Zeroize	N/A

7. Operational Environment

The FIPS 140-2 Area 6 Operational Environment requirements are not applicable because the module does not contain a modifiable operational environment.

8. Security Rules

This section documents the security rules enforced by the BCM5880 Cryptographic Module to implement the security requirements for a FIPS 140-2 Level 3 module.

1. The module shall indicate when an Approved mode of operation is selected.
2. The module implements one Approved mode of operation and one non-FIPS Approved mode of operation. In order to switch between FIPS Approved mode of operation and non-FIPS Approved mode of operation the module must be powered off, the external SECURE_BOOT: CONTROL INPUT: SECURE_BOOT pin moved from high to low, and then the module must be powered on again. Power-cycling zeroizes all volatile plaintext critical security parameters. Module HW\FW\SW enforces that non-volatile plaintext critical security parameters cannot be shared, used, or viewed between modes of operation.
3. Prior to completion of all FIPS power on self-tests the module will perform several special *initialization period* functions (e.g., RAM Memory BIST Read/Write, and BootROM 32-bit Checksum). Failure during these special *initialization period* functions causes a chip reset. Subsequent to the special *initialization period* functions any failure in a FIPS power-on self-test cause the ERROR pin to go high followed by a chip reset.
4. No hardware, software, or firmware components of the cryptographic module are excluded from the security requirements of FIPS 140-2.
5. The module restricts all information flow and physical access points to physical ports and logical interfaces that define all entry and exit points to and from the module.
6. All data output via the data output interface shall be inhibited when an error state exists and during self-tests.
7. The output data path shall be logically disconnected from the circuitry and processes that perform key generation and key zeroization.
8. The module never outputs plaintext cryptographic keys, CSPs, or sensitive data.
9. Status information never contains CSPs or sensitive data that if misused could lead to a compromise of the module
10. The module provides two operator roles. These are the User role and the Cryptographic-Officer role.
11. The module does not support concurrent operators.

12. The module does not support a maintenance role.
13. The module does not support a bypass capability.
14. The module supports identity-based authentication.
15. When the module is powered off and subsequently powered on, the results of previous authentications are not retained and the module requires the operator to be re-authenticated.
16. Authentication data within the module is protected against unauthorized disclosure, modification, and substitution.
17. The module contains the authentication data required to authenticate the operator for the first time.
18. For each attempt to use the authentication mechanism the probability is less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur.
19. For multiple attempts to use the authentication mechanism during a one-minute period the probability is less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.
20. The module's authentication mechanism does not supply any feedback information to the operator.
21. Recovery from "soft" error states is possible via power-cycling. Recovery from "hard" error states is not possible.
22. The module is physically protected with a production-grade hard opaque tamper evident encapsulating material.
23. The module does not contain any doors or removable covers.
24. Secret keys, private keys, and CSPs within the module are protected from unauthorized disclosure, modification, and substitution.
25. Public keys within the module are protected against unauthorized modification and substitution.
26. Cryptographic keys generated by the module are generated using Approved key generation methods: FIP186-2 Appendix 3.1.
27. Compromising the security of the key generation method requires as least as many operations as determining the value of the generated keys.
28. Seed keys are not entered into the module during the key generation process, they are gathered internally.
29. Intermediate key generation values are not output from the module.
30. Key establishment is performed via ECDH SP800-56A (allowed as per FIPS 140-2 Implementation Guidance 7.1).
31. Compromising the security of the key establishment method (2^{128}) requires as many operations as determining the value of the cryptographic key being agreed

- upon (2^{128}).
32. The module does not support manual key entry.
 33. All secret and private keys entered into the module must be encrypted with an ECDH session key.
 34. The module does not support key entry via split knowledge procedures.
 35. The module does not support a SW/FW Load service.
 36. The module provides a method to zeroize all plaintext secret and private cryptographic keys and CSPs within the module (ZEROIZE PIN within the Secure Boot group physical interface turned high).
 37. The module conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B (i.e., for home use).
 38. The module performs the following self-tests:

a. Power up Self-Tests:

- i. Cryptographic algorithm tests:
 - AES [SMAU - Secure Cache] block KAT.
 - AES [SMAU – Generic Crypto/Auth] block KAT.
 - HMAC-SHA-1 & HMAC-SHA-256 [SMAU - Secure Cache] block KAT, covers SHA-1 & SHA-256.
 - HMAC-SHA-1 & HMAC-SHA-256 [SMAU – Generic Crypto/Auth] block KAT, covers SHA-1 & SHA-256.
 - PRNG FIPS 186-2 Appendix 3.1 KAT and Appendix 3.2 KAT.
 - RSA, DSA, ECDSA signature generation and signature verification KAT.
 - SP800-56A ECDH:
 - DLC primitives KAT.
 - Key Agreement KAT.
 - Key Derivation Function KAT.
- ii. Software Integrity Test:
 - BootROM: 32 bit checksum.
 - Secure Boot Image (SBI) the authenticated software extension of the module's Secure Boot Loader, is authenticated by Secure Boot Loader code when Secure Boot Loader code loads the SBI. Authentication is accomplished via 256-bit HMAC verification (the module also decrypts the SBI image with its 128-bit AES key).
- iii. Critical Functions Tests:
 - Memory BIST (Read/Write)

- OTP Checksum Verification

b. Conditional Self-Tests:

- i. Continuous Random Number Generator test – performed on NDRNG and PRNG

39. The operator is capable of commanding the module to perform the power-up self-test via power cycling.

This section documents the security rules imposed by the vendor:

1. The module shall not support the update of the logical serial number or vendor ID.
2. Each 256-bit ECDSA operation takes > 8ms to perform. For each authentication attempt, the cryptographic module has to perform two ECDSA operations, one for ECDSA signature generation and the other for ECDSA signature verification before the operator can be authenticated. The operator can make no more than 3750 attempts in every minute even if attempts were made continuously.

9. Physical Security Policy

Physical Security Mechanisms

The BCM5880 Cryptographic Module includes the following physical security mechanisms:

- Production-grade hard opaque tamper evident encapsulating material.

10. Mitigation of Other Attacks Policy

The module has not been designed to mitigate any specific attacks beyond the requirements of FIPS 140-2.

11. References

- National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-2, January 27, 2000 –
 - <http://csrc.nist.gov/publications/drafts.html>
- National Institute of Standards and Technology, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, Special Publication 800-56A, March 2006.
 - <http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-56-A-1>

12. Definitions and Acronyms

AES: ECB, CBC, CTR, CMAC, CCM	Advanced Encryption Standard as defined by FIPS197 and SP800-38A to SP800-38D
API	Application Programming Interface
BIST	Built-In Self Test
CSP	A FIPS <u>C</u> ritical <u>S</u> ecurity <u>P</u> arameter
DLC	Discrete Logarithm Cryptography
DSA	Digital Signature Algorithm as defined by FIPS186-2
ECDH	Elliptic-curve Diffie-Hellman algorithm
ECDSA	Elliptic-curve Digital Signature Algorithm as defined by FIPS186-2
EMI/EMC	Electromagnetic Interference/Electromagnetic Compatibility
FIPS	Federal Information Processing Standard
FW	Firmware
HMAC	A keyed-Hash Message Authentication Code
HW	Hardware
JTAG	Joint Test Action Group – refer to the test interface standard as defined by IEEE 1149.1 Standard
LPC	Low Pin Count interface
NDRNG	Non Deterministic Random Number Generator
OTP	One Time Programmable memory.
PRNG	Pseudo Random Number Generator
RAM	Random Access Memory
RFID	Pseudorandom Number Generator
ROM	Read Only Memory
RSA	Rivest, Shamir, and Adleman algorithm for public key encryption
SBI	Secure Boot Image. Authenticated software extension of the module's BOOT ROM (note: SBI software is part of the BCM5880 Cryptographic Module).
SCAPI	Simple Cryptographic Application Programming Interface (refer to the crypto library of BCM5880 firmware that utilizes the cryptographic hardware of the BCM5880)

SHA	Secure Hash Algorithm
SMAU	Secure Memory Access Unit
SPI	Synchronous Peripheral Interface
SRAM	Static Random Access Memory
STS TESTING	Statistical Testing
SW	Software
TPM	Trusted Platform Module
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus